MPRA

Munich Personal RePEc Archive

Dynamic Conditioning and Credit Correlation Baskets

Albanese, Claudio and Vidler, Alicia

Independent Consultant, Merrill Lynch

25. January 2008

Combinatorics is an honest subject. No adles, no sigma-algebras. You count balls in a box, and you either have the right number or you havent. Gian Carlo Rota.

DYNAMIC CONDITIONING AND CREDIT CORRELATION BASKETS

CLAUDIO ALBANESE AND ALICIA VIDLER

ABSTRACT. Dynamic conditioning is a technique that allows one to formulate correlation models for large baskets without incurring in the curse of dimensionality. The individual price processes for each reference name can be described by a lattice model specified semi-parametrically or even non-parametrically and which can realistically have about 1000 sites. The time discretization step is chosen so small to satisfy the Courant stability condition and is typically of about a few hours. This constraint ensures needed smoothness for the single name probability kernels which can thus be directly manipulated. A flexible multi-factor correlation model can be obtained by means of conditioning trees corresponding to binomial processes with jumps. There is one conditioning tree associated to each reference names, one associated to each industry sector and a global one to the basket itself. Since the conditioning trees are correlated, the underlying processes are also mutually correlated.

In this paper, we discuss a modeling framework for CDOs based on dynamic conditioning in greater detail than previously done in our other papers. We also show that the model calibrates well to index tranches throughout in the period from 2005 to the Spring of 2008 and yields instructive insights.

CONTENTS

1. Introduction

The quest for a satisfactory modeling framework for bespoke CDOs has attracted considerable interest in recent years. At this time, the pause in new issuance in the structured credit market has altered the business context and also shifted the nature of the modeling problem. As the once established standards for rating and pricing were confronted with severe market volatility, more detailed models that provide a more faithful representation of the economic fundamental drivers have become even more desirable. As we discuss in the last section of this article, applications include not only relative value analysis of illiquid bespoke structures but price dislocations across the capital structure in the index baskets themselves.

We can hardly make justice of the extensive literature in this area by quoting references. A sample of papers which we found stimulating are (Duffie *et al.* 2006), (Duffie *et al.* 2000), (Andersen and Sidenius 2004),(Giesecke and Goldberg 2005), (Hull and White 2003), (Joshi and Stacey 2005), (Lucas *et al.* 2001), (O'Kane and Livesey 2004), (Li 2000), (Schonbucher 2006), (di Graziano and Rogers 2006). The authors first developed models combining jumps and local volatility for single name credit derivatives using analytically solvable schemes (Albanese *et al.* 2003), (Albanese and Chen 2005*a*), (Albanese and Chen 2005*b*), (Albanese and Chen 2004). Then we shifted towards rating models specified semi-parametrically (Albanese *et al.* 2005-2006) and incorporated detailed volatility information in a fully structural, credit-equity model in (Albanese and Vidler 2006).

We concluded that to meet the many objectives of this complex problem, one requires to specify the single name process semi-parametrically and have a flexible correlation framework capturing separately correlations between small moves and correlations between large jumps. The essence of the problem we tackle in this paper, is how to build a model agnostic engineering framework that allows for all the required flexibility. Having accomplished this task, the next is to build an economic model which is as faithful as possible, in the knowledge that modeling assumptions will not affect computational performance.

Dynamic conditioning is a technique that allows one to bypass the curse of dimensionality and build lattice models for complex basket derivatives such as CDOs. We developed the theory over the years and documented early versions of it in (Albanese *et al.* 2005-2006), (Albanese and Vidler 2006) and (Albanese 2006). The version described in this paper is the most advanced we released so far. We give a thorough description of all the equations, sum rules and calibration methods involved in a practical implementation.

Conceptually, dynamic conditioning is not a specific model, but rather a model agnostic engineering framework. The framework allows one to specify single name models flexibly as high dimensional lattice models with stochastic volatility, jumps and all the richness of a generic Markov process one may desire. The vision is to calibrate the single name processes in a pre-processing phase and store intermediate results in the form of conditional probabilities. Single name conditioning is derived

from a super-imposed event tree. The key point is that although the single name processes themselves are technically difficult to correlate, one can still correlate directly the simpler conditioning trees. The conditional probabilities can then be reassembled whenever one needs to price a specific basket, essentially combining the single-name conditional probabilities with each other using combinatorial coefficients which derive out of a correlation model. The correlation model is specified at run-time and is meant to correlate in a fairly arbitrary and flexible way, the single name conditioning trees.

From a mathematical viewpoint, the method is based on the calculation of quarterly transition probability kernels using fast exponentiation, a method based on full matrix multiplications which is equivalent to a direct method with a time discretization step satisfying the Courant condition, i.e. typically as short as a few hours. Choosing a very short time step is essential to ensure sufficient smoothness for the transition probability kernels. See (Albanese 2007) for a proof of convergence of the method in the graph-uniform norm under these conditions, a mathematical result showing how a small time step ensures smoothness. Kernel smoothness is important for applications to dynamic conditioning as modeling correlations requires defining kernel splitting rules, i.e. involves direct manipulation of the transition probability kernels. The numerical robustness of the method is such that all sum-rules are satisfied with relative errors of less than 10^{-10}.

From an engineering viewpoint, the implementation can be divided into a preprocessing stage and a pricing stage. The first involves calculations that depend only on the single name model specification, while the pricing stage depends on the specifications of the correlation model. In the practical example we discuss, the preprocessing stage takes around 138 seconds while the CDO pricing stage takes around 8 seconds on standard hardware with GPU acceleration. The method is numerically very efficient as changes in CDS spreads and correlations do not require a new preprocessing stage, only changes in the underlying equity dynamics do. The method is based on numerical linear algebra and can safely be implemented by multiplying matrices using a single precision engine. As desired, the performance is independent of the model specification.

In this paper, we illustrate applications of our CDO model to datasets from the years 2006-2007, analyzing in the light of the model the changes in the structured credit market that recently took place. We find that the dynamic conditioning model calibrates well to both equity and senior tranches, that it fits the index and satisfies all rigorous sum rules. Interesting insights on the pricing of the mezzanine tranches can be gained out of this analysis. Model parameters are time-homogeneous with only few exceptions such as an exogenous interest rate. We find that a slowly varying inverted term structure for jump correlation is needed for the more recent datasets, not the older ones which calibrate well with time-homogeneous parameters.

To express the correlation model, we condition each single name process by means of a process on a binomial tree with quarterly branchings which also admits jumps to the lowest vertex. See Fig.1 for an illustration. Furthermore, as illustrated in Fig.2, we introduce a binomial tree for each industry sector factor we decide to include and a final binomial tree for the global economy. These trees also have jump transitions to the lowest vertex and they condition each other as illustrated in Fig.2. The conditioning process is described in detail in Section 3. An important aspect
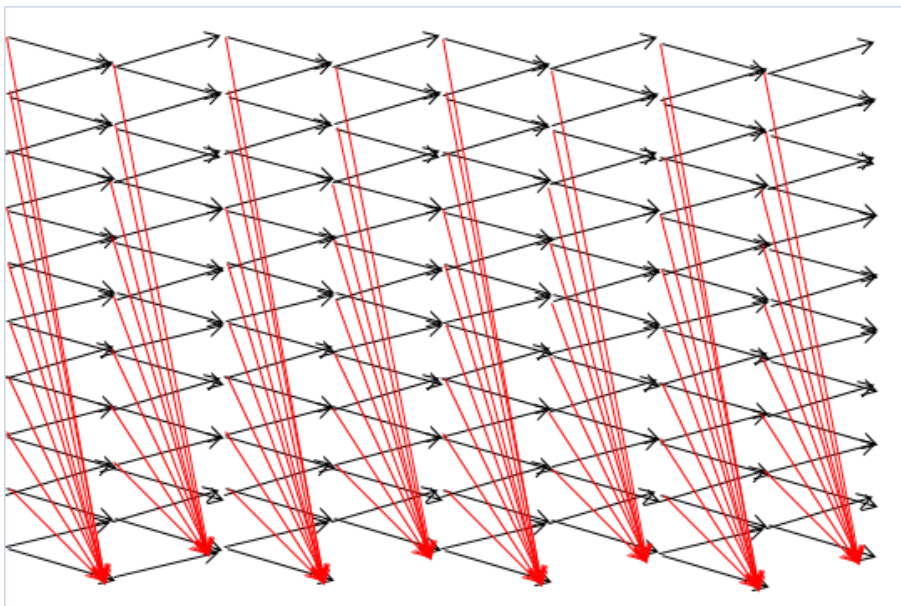
FIGURE 1. Conditioning tree expressing a binomial process with jumps to the bottom vertex and restrictions on the minimum and maximum height achievable.

of the correlation model is that not only the evolution along bonds is correlated, but also jumps to the lowest vertex are correlated. From a calibration standpoint, bond correlations affect to a greater degree the equity tranches while the strength of jump correlation affects to a greater degree the senior and super-senior tranches, thus allowing for a joint calibration at the two opposite sides of the capital structure. Also important is the probability of jump to the lowest vertex in the conditioning lattices: the greater this probability is, the heavier is the weight in the far tail of the loss distribution and the higher are the senior tranche spreads.

In Section 3, we evaluate the marginal and the joint probabilities on the conditioning trees. In Section 4, we discuss how the conditioning process applies to the single name dynamics. Single name calibration involves fitting the initial term structure of CDS spreads for each name. This is achieved by adding jump to default probabilities on the lowest nodes of the single name conditioning trees. The procedure is explained in Section 5. In Section 6, we explain how to evaluate the loss distribution and to price CDO tranches. Finally, in Section 7, we discuss benchmarks and applications to three datasets.

2. The Equity-Driven Single Name Process

The key concept behind model design is that one wants to decouple the economic modeling task from the engineering aspects of pricing. The model itself is specified flexibly in a semi-parametric fashion, resting assured that performance and precision do not depend on model specification. In particular, closed form solvability is never assumed and all calculations are purely numerical.
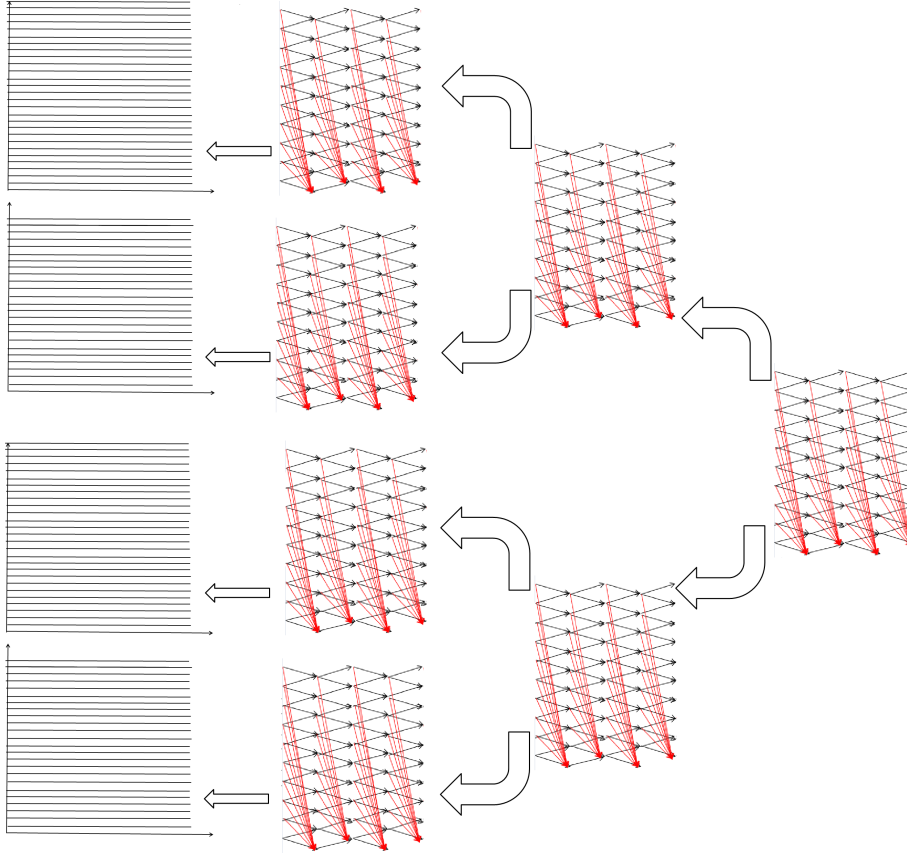
FIGURE 2. Scheme for multifactor conditioning involving industry sector factors and a global economics factor.

Each reference name follows a combination of two processes: an equity driven process defined by means of a Markov generator and a jump to default process overlayed on it in such a way to achieve a precise fit of the term structure of CDS spreads. This section describes the equity-driven process.

Let $T > 0$ be the final maturity and let N_j be the number of periods in which the interval $[0, T]$ is subdivided. Let $j = 0...N_j$ be an integer, let $\Delta T = \frac{T}{N_j}$ and let $T_j = j\Delta T$.

The single name process is defined on a lattice Y labeled by pairs $y = (x, m)$, where $x = 0, 1, ...N_x - 1$ is a variable associated to the single name stock price level and $m = 0, ...N_m - 1$ labels regimes. The points of the form $(x = 0, m)$ for any $m = 0, ...N_m - 1$ are identified with a single state as they all correspond to the state of default, where we impose absorbing boundary conditions. Let $S(x; j)$ be a monotonously increasing function with $S(0; j) = 0$ which gives the equity value corresponding to the state variable x in the time interval $(T_j, T_{j+1}]$.

We make the simplifying assumption that the equity driven process for each reference name is described by the very same process specification. What differentiates one name from another are (i) the initial condition, (ii) the jump to default

process and (iii) the recovery rate. In this Section, we discuss only the Markov generator for the underlying process. In principle, one could use more than one generator if one wanted to model several separate classes of names characterized by processes with different economic characteristics. The impact of such choice would be to increase the preprocessing time by a multiple equal to the number of different dynamic specifications used. The performance of the tranche pricing stage instead would not be affected.

We have described approaches to build and calibrate single name Markov generators in other papers such as (Albanese and Vidler 2006), (Albanese and Osseiran 2007), (Albanese 2006), and we refer to them for more detailed explanations and examples concerning the choice of coefficients. Here we just mention that the Markov generator can often be modeled in the special format

$$(1) \qquad \mathcal{L}(x, m; x', m'; t) = \mathcal{L}(x, x'|m; t)\delta_{mm'} + \mathcal{L}(m, m'|x; t)\delta_{xx'}.$$

The reduced Markovian $\mathcal{L}(x, x'|m; t)$ gives the dynamics in the regime m while the reduced Markovian $\mathcal{L}(m, m'|x; t)$ gives the regime dynamics for a fixed value of the equity state variable x. More general forms are also possible, but this restriction is sufficient for most purposes.

The operator $\mathcal{L}(x, x'|m)$ can be chosen to be of the form of a jump process, i.e.

$$(2) \qquad \mathcal{L}(x, x'|m; t) = \mu(x, m; t)\nabla_j(x, x') - \sigma(x, m; t)\phi(-\Delta_j; t)(x, x')$$

where $t \in (T_j, T_{j+1}]$,

$$(3) \qquad \nabla_j(x, x') = \begin{cases} \frac{\delta_{x+1,x'} - \delta_{x-1,x'}}{S(x+1;j) - S(x-1;j)} & x = 1, ..N_x - 2 \\ 0 & \text{otherwise,} \end{cases}$$

$$(4) \qquad \Delta_j(x, x') = \begin{cases} \frac{\delta_{x+1,x'} + \delta_{x-1,x'} - 2\delta_{x,x'}}{(S(x;j) - S(x-1;j))(S(x+1;j) - S(x;j))} & x = 1, ..N_x - 2 \\ 0 & \text{otherwise,} \end{cases}$$

and $\phi(\lambda)$ is a Bernstein function. The choice $\phi(\lambda) = \lambda$ corresponds to the case of ordinary diffusions and otherwise one obtains a process with jumps. A possible choice for Bernstein corresponds to the gamma subordinator with variance rate $\nu > 0$ given by

$$(5) \qquad \phi_{VG}(\lambda; \nu) = \frac{1}{\nu}\log(1 + \nu\lambda) = \frac{1}{\nu}\int_0^\infty (1 - e^{-\lambda t})t^{-1}e^{-t/\nu}dt.$$

The Markov generator is assumed to depend on time in a piecewise constant fashion. We assume that the Markov generator $\mathcal{L}(x, m; x', m'; t)$ is constant for each $t \in (T_j, T_{j+1}]$ and all $j = 0, ...N_j$. On each interval, we value the period propagator $U(j; y_1, y_2)$ by means of the fast exponentiation method. Namely, let $\delta t_j > 0$ be the largest time interval for which both of the following properties are satisfied:

$$(\text{FE1}) \qquad \min_{y \in \Lambda}(1 + \delta t_j \mathcal{L}(y, y; t)) \geq 1/2$$

$$(\text{FE1}) \qquad \log_2 \frac{\Delta T}{\delta t_j} = n \in \mathbb{N}$$

for $t \in (T_j, T_{j+1}]$ (recall that the Markov generator is modeled as constant as a function of time in each such interval). To compute the propagator

$$(6) \qquad U(j; y_1, y_2) = e^{\Delta T \mathcal{L}(t)}(y_1, y_2; t)$$

for $t \in (T_j, T_{j+1}]$, we first define the elementary propagator

$$(7) \qquad u_{j,\delta t}(y_1, y_2) = \delta_{y_1 y_2} + \delta t \mathcal{L}(y_1, y_2; t)$$

for and then evaluate kernels iteratively, as follows:

$$u_{j,2\delta t} = u_{j,\delta t} \cdot u_{j,\delta t},$$
$$u_{j,4\delta t} = u_{j,2\delta t} \cdot u_{j,2\delta t},$$
$$....$$
$$(8) \qquad u_{j,2^n \delta t} = u_{j,2^{n-1}\delta t} \cdot u_{j,2^{n-1}\delta t}.$$

3. THE CONDITIONING TREES

The conditioning trees are binomial trees with jumps to the lowest node. In this Section, we define the tree dynamics and show how to correlate two trees and find joint probability distributions.

3.1. Tree geometry.

We build a binomial tree which branches off at the time nodes T_j, see Fig. 1. The associated stochastic process describes a random walk starting from the root vertex and visiting vertices at later times. As the conditioning state variable evolves from a vertex at time T_j on to a vertex at time T_{j+1}, the process can either follow a bond originating from the starting vertex or jump to the lowest vertex in the tree. The tree itself is not a full binomial tree but it is truncated at the bottom and at the top for the sole purpose to economize memory and CPU time by cutting off states of the world that could be reached only with very small probability.

More precisely, consider the untruncated binomial tree in the period $(T_j, T_{j+1}]$. Let $N_n(j)$ be the number of bonds in this interval and let $N_v(j)$ be the number of vertices at time T_j. If b is a bond in the time interval $(T_j, T_{j+1}]$, let $V_\rightarrow(j, b)$ be the vertex at time T_{j+1} where the bond b terminates and let $V_\leftarrow(j, b)$ be the vertex at time T_j where the bond b originates. Let us define $\mathtt{dir}(b) = \nearrow$ if the bond b is directed upwards while $\mathtt{dir}(b) = \searrow$ if the bond b is directed downwards.

To define the truncated tree we introduce three integers $h_{\min} < h_0 < h_{\max}$. Let $h(v, j)$ be the height of the vertex v at time T_j. This function is defined recursively so that the height of the root vertex is given by $h(v, 0) = h_0$ and so that if b is a bond in the interval $(T_j, T_{j+1}]$, then

$$(9) \qquad h(V_\rightarrow(j, b), j) = h(V_\leftarrow(j, b), j) + \delta_{\mathtt{dir}(b), \nearrow} - \delta_{\mathtt{dir}(b), \searrow}.$$

The truncated tree is defined as the subtree comprising all vertices of height such that $h_{\min} < h(v, j) < h_{\max}$ and all the bonds joining pairs of such vertices. Let $b_{\min}(j)$ be the index of the lowest bond in the interval $(T_j, T_{j+1}]$ in the truncated tree and let $b_{\max}(j)$ be the highest bond. Similarly, let $v_{\min}(j)$ be the index of the lowest vertex at time T_j and let $v_{\max}(j)$ be the highest vertex. There is some freedom on how to define these indices as the offset is arbitrary.

3.2. Tree parameters.

The parameters of the correlation model can be listed as follows:

- $h_{\min}$: the minimum height;
- $h_{\max}$: the maximum height;
- $q(\nearrow)$: the probability of an up move in the conditioning tree;

- $q(\searrow)$: the probability of a down move along a bond in the conditioning tree;
- $q(\downarrow)$: the jump probability;
- h_0: the height of the root vertex;
- $\rho_{ab}(a, j)$: the bond correlation term structure for the name a conditioning process and the conditioning process for the corresponding sector $\texttt{sec}(a)$;
- $\rho_{\texttt{sec}b}(\texttt{sec}, j)$: the bond correlation term structure between the conditioning process for sector $\texttt{sec}$ and the global conditioning process;
- $\rho_{aj}(a, j)$: the jump correlation term structure for the conditioning process relative to the reference name a and the conditioning process for the corresponding sector;
- $\rho_{\texttt{sec}j}(a, j)$: the jump correlation term structure for the conditioning process relative to the sector $\texttt{sec}$ and the global conditioning process.

In our example, we make the simplifying assumption that there exist two functions $\rho_b(j)$ and $\rho_j(j)$ such that

$$(10) \qquad\qquad \rho_{ab}(a, j) = \rho_{\texttt{sec}b}(\texttt{sec}, j) = \rho_b(j)$$

and

$$(11) \qquad\qquad \rho_{aj}(a, j) = \rho_{\texttt{sec}j}(\texttt{sec}, j) = \rho_j(j)$$

for all reference names a and all sectors $\texttt{sec}$.

Let $\pi_b(j, b)$ be the conditional probability for a transition to the bond b, conditioned to knowing that the starting vertex is $V_\leftarrow(j, b)$. Similarly, let $\pi_\downarrow(j, v)$ be the conditional probability of a jump in the tree conditional to starting from the vertex v at time T_j. Let b be a bond and suppose that a total of two bonds originate from the vertex $v = V_\leftarrow(j, b)$. In this case, we set

$$(12) \qquad \pi_b(j, b) = q(\texttt{dir}(b)) \qquad \text{and} \qquad \pi_\downarrow(j, v) = q(\downarrow).$$

If instead only the bond b originates from the starting vertex $v = V_\leftarrow(j, b)$, then we set

$$(13) \qquad\qquad \pi_b(j, b) = 1 - q(\downarrow), \qquad \pi_\downarrow(j, v) = q(\downarrow).$$

3.3. **Marginal Probabilities.** Let us introduce the following notations:

- $P_b(j, b)$ is the occupation probability for the bond b in the time interval $j = 0..N_j - 1$.
- $P_\downarrow(j, v)$ is the probability that a jump occurs from the vertex v at time T_j to the bottom vertex $v_{\min}(j + 1)$ at time T_{j+1}.
- $P_v(j, v)$ is the probability that the vertex v at time T_j is occupied.

If $j = 0$, we have that

$$(14) \qquad\qquad P_v(j, 0) = 1$$

$$(15) \qquad\qquad P_b(j, 0) = \pi_b(j, 0)$$

$$(16) \qquad\qquad P_b(j, 1) = \pi_b(j, 1)$$

$$(17) \qquad\qquad P_\downarrow(j, 0) = \pi_\downarrow(j, 0).$$

Next, suppose that $j = 1, ..N_j$, let b_1 be a bond in the interval $(T_j, T_{j+1}]$ and let $v = V_\rightarrow(j, b_1)$. Assume first that $v > v_{\min}(j)$, i.e. v is not a lowest vertex. Then, if there exists a second bond b_2 which also terminates at v, we set

$$(18) \qquad\qquad P_v(j, v) = P_b(j - 1, b_1) + P_b(j - 1, b_2).$$

Otherwise, if b_1 is the single bond terminating at v, then

$$(19) \qquad P_v(j, v) = P_b(j - 1, b_1).$$

If $v = v_{\min}(j)$ instead, these formulas read as follows:

$$(20) \qquad P_v(j, v_{\min}(j)) = P_b(j - 1, b_1) + P_b(j - 1, b_2) + \sum_{v_1 = v_{\min}(j-1)}^{v_{\max}(j-1)} P_\downarrow(j - 1, v_1)$$

in case two bonds emerge from v, otherwise

$$(21) \qquad P_v(j, v_{\min}(j)) = P_b(j - 1, b_1) + \sum_{v_1 = v_{\min}(j-1)}^{v_{\max}(j-1)} P_\downarrow(j - 1, v_1).$$

If $j = 1, .. N_j - 1$, the bond occupation probabilities are given by

$$(22) \qquad P_b(j, b) = \pi_b(j, b) \cdot P_v(j, V_\leftarrow(j, b))$$

and the jump occurrence probabilities are

$$(23) \qquad P_\downarrow(j, v) = \pi_\downarrow(j, v) \cdot P_v(j, v).$$

3.4. **Conditional Joint Probabilities.** In the following, we consider two correlated trees. The discussion is general and applies to all situations of interest in the CDO model. For instance, the first tree could be a single name conditioning tree and the second could be a sector tree. Or the first tree could be a sector tree and the second tree the global conditioning tree. In the following set of definitions, we denote with the subscript 1 a bond or vertex in the first tree and with the subscript 2 a bond or vertex in the second tree. Also, bonds referred b_1, b_2 to in the paragraph below are both in the interval $(T_j, T_{j+1}]$ while vertices v_1, v_2 are both at time T_j.

- $\pi_{bb}(b_1, b_2)$ is the conditional joint probability for the pair of bonds (b_1, b_2) in the interval $(T_j, T_{j+1}]$, conditional to the pair of vertices $(V_\leftarrow(j, b_1), V_\leftarrow(j, b_2))$ being visited at time T_j;
- $\pi_{jb}(j, v_1, b_2)$ is the conditional joint probability for the vertex v_1 in the first tree being visited at time T_j and a jump occurring from there to the lowest node while the bond b_2 is visited on the second tree. This probability is conditional to the event that the vertex $V_\leftarrow(j, b_2))$ on the second tree is also visited at time T_j;
- $\pi_{bj}(j, b_1, v_2)$ is the conditional joint probability for the vertex v_2 in the second tree being visited at time T_j and a jump occurring from there to the lowest node while the bond b_1 is visited on the first tree. This probability is conditional to the event that the vertex $V_\leftarrow(j, b_1))$ on the first tree is also visited at time T_j;
- $\pi_{jj}(j, v_1, v_2)$ is the conditional joint probability for the vertex v_1 in the first tree being visited at time T_j and a jump occurring from there to the lowest node while the vertex v_2 in the second tree is also visited and a jump also occurs from there to the lowest node.

In the following, a bond b is called a *single bond* if it is the only bond emerging from the vertex $V_\leftarrow(b)$. The bond b is called *twin bond* if there are two bonds originating from the vertex $V_\leftarrow(b)$.

Assuming that transition probabilities are height-independent in the tree, these conditional joint probabilities are all given in terms of the following constants:

- $q(\searrow, \searrow) = \pi_{bb}(b_1, b_2)$ in case b_1 and b_2 are both twin bonds and they both go down;
- $q(\searrow, \nearrow) = \pi_{bb}(b_1, b_2)$ in case b_1 and b_2 are both twin bonds, b_1 goes down and b_2 goes up;
- $q(\searrow, \downarrow) = \pi_{bj}(b_1, v_2)$ in case b_1 is a twin bond going down and v_2 is a vertex;
- $q(\downarrow, \downarrow) = \pi_{jj}(v_1, v_2)$ where v_1 and v_2 are vertices;
- $q(\nearrow, \nearrow) = \pi_{bb}(b_1, b_2)$ in case b_1 and b_2 are both twin bonds and they both go up;
- $q(\nearrow, \downarrow) = \pi_{bj}(b_1, b_2)$ in case b_1 is a twin bond going up and v_2 is a vertex;
- $q(\searrow, \rightarrow) = \pi_{bb}(b_1, b_2)$ in case b_1 is a twin bond going down and b_2 is a single bond;
- $q(\nearrow, \rightarrow) = \pi_{bb}(b_1, b_2)$ in case b_1 is a twin bond going up and b_2 is a single bond;
- $q(\rightarrow, \downarrow) = \pi_{bj}(b_1, v_2)$ in case b_1 is a single bond and v_2 is a vertex;
- $q(\rightarrow, \rightarrow) = \pi_{bb}(b_1, b_2)$ in case b_1 and b_2 are both single bonds.

We have that

$$
\begin{aligned}
q(\searrow, \searrow) =&(1 - \rho_j(j)) \cdot (1 - \rho_b(j)) \cdot q(\searrow)^2 + (1 - \rho_j(j)) \cdot \rho_b(j) \cdot \frac{q(\searrow)^2}{q(\searrow) + q(\downarrow)} \\
&+ \rho_j(j) \cdot (1 - \rho_b(j)) \cdot \frac{q(\searrow)^2}{1 - q(\downarrow)} + \rho_j(j) \cdot \rho_b(j) \cdot q(\searrow)
\end{aligned}
$$

(24)

$$
\begin{aligned}
q(\nearrow, \nearrow) =&(1 - \rho_j(j)) \cdot (1 - \rho_b(j)) \cdot q(\nearrow)^2 + (1 - \rho_j(j)) \cdot \rho_b(j) \cdot \frac{q(\nearrow)^2}{q(\nearrow) + q(\downarrow)} \\
&+ \rho_j(j) \cdot (1 - \rho_b(j)) \cdot \frac{q(\nearrow)^2}{1 - q(\downarrow)} + \rho_j(j) \cdot \rho_b(j) \cdot q(\nearrow),
\end{aligned}
$$

(25)

$$
q(\searrow, \nearrow) =(1 - \rho_j(j)) \cdot (1 - \rho_b(j)) \cdot q(\searrow)q(\nearrow), + \rho_j(j) \cdot (1 - \rho_b(j)) \cdot \frac{q(\nearrow) \cdot q(\searrow)}{1 - q(\downarrow)},
$$

(26)

$$
q(\searrow, \downarrow) =(1 - \rho_j(j)) \cdot (1 - \rho_b(j)) \cdot q(\searrow) \cdot q(\downarrow), + (1 - \rho_j(j)) \cdot \rho_b(j) \cdot \frac{q(\searrow) \cdot q(\downarrow)}{q(\searrow) + q(\downarrow)},
$$

(27)

$$
q(\nearrow, \downarrow) =(1 - \rho_j(j)) \cdot (1 - \rho_b(j)) \cdot q(\nearrow) \cdot q(\downarrow), + (1 - \rho_j(j)) \cdot \rho_b(j) \cdot \frac{q(\nearrow) \cdot q(\downarrow)}{q(\nearrow) + q(\downarrow)},
$$

(28)

$$
\begin{aligned}
q(\downarrow, \downarrow) =&(1 - \rho_j(j)) \cdot (1 - \rho_b(j)) \cdot q(\downarrow)^2, \\
&+ (1 - \rho_j(j)) \cdot \rho_b(j) \cdot q(\downarrow) \cdot \left(1 - \frac{q(\searrow)}{q(\searrow) + q(\downarrow)} - \frac{q(\nearrow)}{q(\nearrow) + q(\downarrow)}\right), \\
&+ \rho_j(j) \cdot (1 - \rho_b(j)) \cdot q(\downarrow) + + \rho_j(j) \cdot \rho_b(j) \cdot q(\downarrow),
\end{aligned}
$$

(29)
$$
q(\searrow, \rightarrow) = q(\searrow) - q(\searrow, \downarrow),
$$

$$(30) \qquad q(\rightarrow, \downarrow) = q(\downarrow) - q(\downarrow, \downarrow),$$

$$(31) \qquad q(\nearrow, \rightarrow) = q(\nearrow) - q(\nearrow, \downarrow),$$

$$(32) \qquad q(\rightarrow, \rightarrow) = q(\searrow) + q(\nearrow) - q(\rightarrow, \downarrow).$$

These joint probabilities satisfy the following constraints

$$(33) \qquad q(\searrow, \searrow) + q(\searrow, \nearrow) + q(\searrow, \downarrow) = q(\searrow)$$

$$(34) \qquad q(\searrow, \nearrow) + q(\nearrow, \nearrow) + q(\nearrow, \downarrow) = q(\nearrow)$$

$$(35) \qquad q(\searrow, \downarrow) + q(\nearrow, \downarrow) + q(\downarrow, \downarrow) = q(\downarrow)$$

$$(36) \quad q(\searrow, \searrow) + q(\nearrow, \nearrow) + 2 \cdot q(\searrow, \nearrow) + 2 \cdot q(\searrow, \downarrow) + 2 \cdot q(\nearrow, \downarrow) + q(\downarrow, \downarrow) = 1,$$

$$(37) \qquad q(\searrow, \rightarrow) + q(\searrow, \downarrow) = q(\searrow),$$

$$(38) \qquad q(\rightarrow, \downarrow) + q(\downarrow, \downarrow) = q(\downarrow),$$

$$(39) \qquad q(\searrow, \rightarrow) + q(\nearrow, \rightarrow) + q(\rightarrow, \downarrow) = P_b,$$

$$(40) \qquad q(\nearrow, \rightarrow) + q(\nearrow, \downarrow) = q(\nearrow),$$

$$(41) \qquad q(\searrow, \downarrow) + q(\nearrow, \downarrow) + q(\downarrow, \downarrow) = q(\downarrow),$$

$$(42) \qquad q(\searrow, \rightarrow) + q(\nearrow, \rightarrow) + q(\rightarrow, \downarrow) + q(\searrow, \downarrow) + q(\nearrow, \downarrow) + q(\downarrow, \downarrow) = 1,$$

$$(43) \qquad q(\rightarrow, \rightarrow) + q(\rightarrow, \downarrow) = P_b,$$

$$(44) \qquad q(\rightarrow, \rightarrow) + 2q(\rightarrow, \downarrow) + q(\downarrow, \downarrow) = 1.$$

3.5. **(Unconditional) Joint Probabilities.** Let us introduce the following notations:

- $P_{bb}(j, b_1, b_2)$ is the joint probability for the pair of bonds (b_1, b_2) being visited in the period $[T_j, T_{j+1})$;
- $P_{vv}(j, v_1, v_2)$ is the joint probability for the pair of vertices (v_1, v_2) being visited at time T_j;
- $P_{jj}(j, v_1, v_2)$ is the joint probability for the pair of vertices (v_1, v_2) being visited at time T_j followed in both cases by jumps to the lowest node;
- $P_{jb}(j, v_1, b_2)$ is the joint probability for the vertex v_1 on the first tree being visited at time T_j followed by a jump to the lowest node, while the bond b_2 is visited in the time interval $(T_j, T_{j+1}]$ in the second tree;
- $P_{bj}(j, b_1, v_2)$ is the joint probability for the vertex v_2 in the second tree being visited at time T_j followed by a jump to the lowest node, while the bond b_1 is visited in the time interval $(T_j, T_{j+1}]$ on the first tree.

The determination of these joint probabilities proceeds by induction in j. If $j = 0$, we set

$$P_{vv}(0, 0, 0) = 1. \tag{45}$$

Iterating over the bonds, if $j = 0, ...N_j - 1$ we find

$$P_{bb}(j, b_1, b_2) = P_{vv}(j, V_{\leftarrow}(j, b_1), V_{\leftarrow}(j, b_2)) \cdot \pi_{bb}(j, b_1, b_2) \tag{46}$$

$$P_{bj}(j, b_1, v_2) = P_{vv}(j, V_{\leftarrow}(j, b_1), v_2) \cdot \pi_{bj}(j, b_1, v_2) \tag{47}$$

$$P_{bj}(j, b_1, v_2) = P_{vv}(j, V_{\leftarrow}(j, b_1), v_2) \cdot \pi_{bj}(j, b_1, v_2) \tag{48}$$

$$P_{jj}(j, v_1, v_2) = P_{vv}(j, v_1, v_2) \cdot \pi_{jj}(j, v_1, v_2). \tag{49}$$

Next, assume that $j = 1, ...N_j$ and consider the joint probability $P_{vv}(j, v_1, v_2)$ where (v_1, v_2) is a pair of vertices at time T_j. We have that

$$P_{vv}(j, v_1, v_2) =$$

$$\sum_{\substack{b_1 : v_1 = V_{\rightarrow}(j-1, b_1) \\ b_2 : v_2 = V_{\rightarrow}(j-1, b_2)}} P_{vv}(j-1, V_{\leftarrow}(j-1, b_1), V_{\leftarrow}(j-1, b_2)) \cdot \pi_{bb}(j-1, b_1, b_2)$$

$$+ \delta_{v_1, v_{\min}(j-1)} \sum_{\substack{w_1 = v_{\min}(j-1)..v_{\max}(j-1) \\ b_2 : v_2 = V_{\rightarrow}(j-1, b_2)}} P_{vv}(j-1, w_1, V_{\leftarrow}(j-1, b_2)) \cdot \pi_{jb}(j-1, w_1, b_2)$$

$$+ \delta_{v_2, v_{\min}(j-1)} \sum_{\substack{b_1 : v_1 = V_{\rightarrow}(j-1, b_1) \\ w_2 = v_{\min}(j-1)..v_{\max}(j-1)}} P_{vv}(j-1, V_{\leftarrow}(j-1, b_1), w_2) \cdot \pi_{bj}(j-1, b_1, w_2)$$

$$+ \delta_{v_1, v_{\min}(j-1)} \delta_{w_2, v_{\min}(j-1)} \sum_{\substack{w_1 = v_{\min}(j-1)..v_{\max}(j-1) \\ w_2 = v_{\min}(j-1)..v_{\max}(j-1)}} P_{vv}(j-1, w_1, w_2) \cdot \pi_{jj}(j-1, w_1, w_2).$$

(50)

4. Dynamic Conditioning

As a first step, one needs to define an order relationship $\prec$ in the state space Y. This can be done using for instance the value of 5 year CDS spreads. Having done that, one performs an iterative construction in j starting from $j = 0$ and constructing all bond conditioned quantities of interest up to $j = N_j - 1$ and all vertex conditioned quantities up to $j = N_j$.

Let us introduce the following notations:

- $U(j; y_1, y_2)$ is the propagator in the time interval $(T_j, T_{j+1}]$;
- $U(j, \searrow; y_1, y_2)$ is the conditional propagator in the time interval $(T_j, T_{j+1}]$ in case the single name conditioning process visits a bond going down;
- $U(j, \nearrow; y_1, y_2)$ is the conditional propagator in the time interval $(T_j, T_{j+1}]$ in case the single name conditioning process visits a bond going up;
- $U(j, \downarrow; y_1, y_2)$ is the conditional propagator in the time interval $(T_j, T_{j+1}]$ in case the single name conditioning process jumps to the bottom vertex;

- $U(j, \rightarrow; y_1, y_2)$ is the conditional propagator in the time interval $(T_j, T_{j+1}]$ in case the single name conditioning follows a bond, i.e. does not jump;
- $D_a^{\mathrm{eq}}(y, j, b)$ is the probability of equity driven default in the time interval $(T_j, T_{j+1}]$ conditional to the process starting from y at time 0 and to the single name conditioning process to visit the bond b in the same time interval;
- $D_{by}^{\mathrm{eqb}}(y, j, b)$ is the probability of equity driven default in the time interval $(T_j, T_{j+1}]$ conditional to the process starting from y at time 0 and to the single name conditioning process visiting the bond b in the same time interval;
- $D_{vy}^{\mathrm{eqj}}(y, j, v)$ is the probability of equity driven default in the time interval $(T_j, T_{j+1}]$ conditional to the process starting from y at time 0 and to the single name conditioning process visiting the vertex v at time T_j and then jumping to the lowest vertex in the subsequent time interval;
- $\overline{D_{vy}^{\mathrm{eq}}}(y, j, v)$ is the cumulative probability of equity driven default at time T_j conditional to the process starting from y at time 0 and to the single name conditioning process to visit the vertex v at time T_j;
- $D_y^{\mathrm{eq}}(y, j)$ is the probability of equity driven default in the time interval $(T_j, T_{j+1}]$ conditional to the process starting from y at time 0;
- $\overline{D_y^{\mathrm{eq}}}(y, j, v)$ is the cumulative probability of equity driven default up to time T_j conditional to the process starting from y at time 0;
- $U_v(v; y_1, y_2)$ is conditional the propagator in the time interval $(T_0, T_j]$ in case the single name conditioning process visits the vertex v at time T_j;
- $U_b(b; y_1, y_2)$ is conditional the propagator in the time interval $(T_0, T_{j+1}]$ in case the single name conditioning process visits the bond b in the time interval $(T_j, T_{j+1}]$;
- $U_\downarrow(v; y_1, y_2)$ is conditional the propagator in the time interval $(T_0, T_{j+1}]$ in case the single name conditioning process visits the vertex v at time T_j and from the jumps to the lowest vertex in the time interval $(T_j, T_{j+1}]$.

4.1. **Propagators with Local Conditioning.** Let $j = 0, ..N_j - 1$. The first step is to build the unconditional kernel $U(j; y_1, y_2)$ by fast exponentiation, as explained in Section 2. Having done that, one finds the kernels $U(j, \downarrow; y_1, y_2)$, $U(j, \searrow; y_1, y_2)$ and $U(j, \nearrow; y_1, y_2)$ in sequence. Let us fix $y_1 \in Y$. Let

$$(51) \qquad \xi_\downarrow = \inf_{\prec} \left\{ y_3 \in Y : \sum_{y_2 \prec y_3} U(j; y_1, y_2) \geq q(\downarrow) \right\}.$$

We set

$$(52) \qquad U(j, \downarrow; (y_1, y_2)) = q(\downarrow)^{-1} U(j; y_1, y_2)$$

for all $y_2 \prec \xi_\downarrow$. Furthermore

$$(53) \qquad U(j, \downarrow; y_1, \xi_\downarrow) = 1 - \sum_{y_2 \prec \xi_\downarrow} U(j, \downarrow; y_1, y_2)$$

and otherwise $U(j, \downarrow; y_1, y_2) = 0$ for all $y_2 \succ \xi_\downarrow$.

We then set $U(j, \searrow; (y_1, y_2)) = 0$ for all $y_2 \prec \xi_\downarrow$ and

$$(54) \qquad U(j, \searrow; y_1, \xi_\downarrow) = U(j; y_1, \xi_\downarrow) - U(j, \downarrow; y_1, \xi_\downarrow).$$

Next, let

$$(55) \qquad \xi_\searrow = \inf_\prec \left\{ y_3 \in Y : U(j, \searrow; y_1, \xi_\downarrow) + \sum_{\substack{y_2 \prec y_3 \\ y_2 \succ \xi_\downarrow}} U(j; y_1, y_2) \geq q(\downarrow) + q(\searrow) \right\}.$$

and set

$$(56) \qquad U(j, \searrow; y_1, y2) = q(\searrow)^{-1} U(j; y_1, y_2)$$

for all $y_2 \prec \xi_\searrow$ and

$$(57) \qquad U(j, \searrow; y_1, \xi_\searrow) = 1 - \sum_{y_2 \prec \xi_\searrow} U(j, \searrow; y_1, y_2).$$

Also $U(j, \searrow; y_1, y_2) = 0$ for all $y_2 \succ \xi_\searrow$.

Finally, we set $U(j, \nearrow; y_1, y_2) = 0$ for all $y_2 \prec \xi_\searrow$,

$$(58) \qquad U(j, \nearrow; y_1, \xi_\searrow) = U(j; y_1, \xi_\downarrow) - U(j, \searrow; y_1, \xi_\searrow)$$

and

$$(59) \qquad U(j, \nearrow; y_1, y_2) = q(\nearrow)^{-1} U(j; y_1, y_2)$$

for all $y_2 \succ \xi_\searrow$.

We also require the propagator on the interval $(T_j, T_{j+1}]$ conditional not to have a jump to the lowest vertex in the single name conditioning process, i.e.

$$(60) \qquad U(j, \rightarrow; y_1, y_2) = \frac{q(\nearrow) \cdot U(j, \nearrow; y_1, y_2) + q(\searrow) \cdot U(j, \searrow; y_1, y_2)}{q(\nearrow) + q(\searrow)}.$$

4.2. Propagators with Global Conditioning. If $j = 0$ we set

$$(61) \qquad \overline{D_{vy}^{\mathrm{eq}}}(j = 0; v, y_1) = 0$$

$$(62) \qquad U_v(j = 0; y_1, y_2) = \delta_{y_1, y_2}.$$

Let $j > 0$ and let v be a vertex at time T_j. Suppose first that $v > v_{\min}(j)$. Let b_1 be the bond with least index such that $V_\rightarrow(b_1) = v$. The existence of a second bond b_2 such that $V_\rightarrow(b_2) = v$ is not granted, but in the case such a bond exists, let us define

$$(63) \qquad p_1 = \frac{P_b(j - 1, b_1)}{P_b(j - 1, b_1) + P_b(j - 1, b_2)}$$

$$(64) \qquad p_2 = \frac{P_b(j - 1, b_2)}{P_b(j - 1, b_1) + P_b(j - 1, b_2)}.$$

Otherwise, if a second bond b_2 such that $V_\rightarrow(b_2) = v$ does not exist, we set

$$(65) \qquad p_1 = 1 \qquad p_2 = 0.$$

If $v > v_{\min}(j)$, then

$$(66) \qquad U_v(j, v, y_1, y_2) = p_1 U_b(j - 1, b_1, y_1, y_2) + p_2 U_b(j - 1, b_2, y_1, y_2).$$

Next, suppose first that $v = v_{\min}(j)$. Let b_1 be the bond with least index such that $V_\rightarrow(b_1) = v$. Again he existence of a second bond b_2 such that $V_\rightarrow(b_2) = v$ is not granted, but in case such a bond exists, let us define

$$(67) \qquad Z = P_b(j - 1, b_1) + P_b(j - 1, b_2) + \sum_{v_0 = v_{\min}(j-1)}^{v_{\max}(j-1)} P_\downarrow(j - 1, v_0)$$

Otherwise, we set

$$\text{(68)} \qquad Z = P_b(j-1, b_1) + \sum_{v_0=v_{\min}(j-1)}^{v_{\max}(j-1)} P_\downarrow(j-1, v_0).$$

As one proceeds with the iterative construction of conditional propagators, the bond conditional propagators need to be evaluated after obtaining the vertex conditioned ones. If $j = 0$, we set

$$\text{(69)} \qquad U_\downarrow y_1, y_2) = U(j, \downarrow; y_1, y_2)$$

$$\text{(70)} \qquad U_b y_1, y_2) = \delta_{\texttt{dir}(b), \searrow} U(j, \searrow; y_1, y_2) + \delta_{\texttt{dir}(b), \nearrow} U(j, \nearrow; y_1, y_2).$$

Next consider the case $j > 0$. Let b be a bond. If b is the only bond starting from the vertex $V_\leftarrow(j, b)$, then we set

$$\text{(71)} \qquad U_b(j, b, y_1, y_2) = \sum_{y_3} U_v(j, V_\leftarrow(j, b), y_1, y_3) U(j, \rightarrow; y_3, y_2).$$

Otherwise, we set

$$\text{(72)} \qquad U_b(j, b, y_1, y_2) = \sum_{y_3} U_v(j, V_\leftarrow(j, b), y_1, y_3) K(j, \texttt{dir}(b), y_3, y_2).$$

Finally

$$\text{(73)} \qquad U_\downarrow(j, v, y_1, y_2) = \sum_{y_3} U_v(j, V_\leftarrow(j, b), y_1, y_3) U(j, \downarrow; y_3, y_2).$$

The bond-conditional probabilities of default are

$$\text{(74)} \qquad D_{by}^{\text{eqb}}(y_1, j, b) = U_b(b, j, y_1, 0) - \overline{D_{vy}^{\text{eq}}}(y_1, j, V_\leftarrow(b))$$

while the jump-conditional probabilities of default are

$$\text{(75)} \qquad D_{vy}^{\text{eqj}}(y_1, j, v) = U_\downarrow(v, j, y_1, 0) - \overline{D_{vy}^{\text{eq}}}(y_1, j, v).$$

Finally

$$\text{(76)} \qquad D_y^{\text{eq}}(y_1, j) = \sum_{b=b_{\min}}^{b_{\max}} P_b(j, b) D_{by}^{\text{eqb}}(y_1, j, b) + \sum_{v=v_{\min}(j)}^{v_{\max}(j)} P_v(j, v) D_{vy}^{\text{eqj}}(y_1, j, v).$$

5. SINGLE NAME CALIBRATION

The model needs to be calibrated based on individual CDS spread information and other inputs. Bootstrapping a CDS curve and finding the implied default probabilities based on recovery assumptions are standard tasks over which we do not dwell here. We just assume that we are given the following inputs:

- $Z(T_j)$ the discount factor for the maturity T_j.
- N_a is the number of reference names;
- $R(a, j)$ is the input expected recovery rate for name a conditional to default occurring in the time interval $(T_j, T_{j+1}]$;
- $\lambda_a(j)$ is the implied probability of default in the time interval $(T_j, T_{j+1}]$ conditional to know default occurring prior to that;
- $y_0(a)$ is the initial condition for the reference name a.

Defaults in our model correspond to events whereby the stock price variable $S(x_t) = 0$ at some time t. Defaults can either occur because the x_t process as described above hits the zero boundary. This events are referred to as *equity-driven defaults*. As an alternative, in order to calibrate the single name dynamics, we allow for jump to defaults to take place whenever the single name conditioning process is visiting the lowest bonds. The bond-dependent probability of jump to default is adjusted in such a way to precisely fit the term structure of CDS spreads. To this end, when initializing the state variable for each individual name, one needs to ensure that the probabilities of equity driven default corresponding to this choice are strictly bounded from above by the implied probabilities of default as derived from market CDS spread curves. The procedure to calibrate such probabilities is given in the remainder of this Section.

We define and value the following functions:

- $D_{ba}^{\text{eqb}}(a, j, b)$ is the probability of equity driven default in the time interval $(T_j, T_{j+1}]$ conditional to the process starting from $y_0(a)$ at time 0 and to the single name conditioning process visiting the bond b in the same time interval;

- $D_{va}^{\text{eqj}}(a, j, v)$ is the probability of equity driven default in the time interval $(T_j, T_{j+1}]$ conditional to the process starting from $y_0(a)$ at time 0 and to the single name conditioning process visiting the vertex v at time T_j and then jumping to the lowest vertex in the subsequent time interval;

- $\overline{D_{va}^{\text{eq}}}(a, j, v)$ is the cumulative probability of equity driven default at time T_j conditional to the process starting from $y_0(a)$ at time 0 and to the single name conditioning process to visit the vertex v at time T_j;

- $\overline{D_{va}}(a, j, v)$ is the conditional cumulative probability that reference name a defaults by time T_j, conditional to the single name conditioning process visiting the vertex v at time T_j;

- $\overline{D_a}(a, j)$ is the cumulative probability that reference name a defaults by time T_j;

- $D_{ba}^{\text{jtd}}(a, j, b)$ is the conditional probability that a jump-to-default event for reference name a occurs in the time interval $(T_j, T_{j+1}]$, conditioned to no default occurring prior to time T_j and conditioned to the single name process visiting the bond b in the same time interval;

- $D_a^{\text{jtd}}(a, j)$ is the probability that a jump-to-default event for reference name a occurs in the time interval $(T_j, T_{j+1}]$ and no default occurs prior to time T_j;

- $D_{ba}(a, j, b)$ is the conditional probability that a default event for reference name a occurs in the time interval $(T_j, T_{j+1}]$ and no default occurring prior to time T_j, conditioned to the single name process visiting the bond b in the same time interval;

- $D_{ja}(a, j, v)$ is the conditional probability that a default event for reference name a occurs in the time interval $(T_j, T_{j+1}]$ and no default occurring prior to time T_j, conditioned to the single name process visiting the vertex v at time T_j and then jumping to the lowest vertex immediately thereafter;

- $D_a(a, j)$ is the conditional probability that a default event for reference name a occurs in the time interval $(T_j, T_{j+1}]$ and no default occurs prior to time T_j;

- $\pi_{ba}^{\text{jtd}}(a, j, b)$ is the conditional probability that a default event for reference name a occurs in the time interval $(T_j, T_{j+1}]$ conditional to no default occurring prior to time T_j and to the conditioning process visiting the bond b in the same time interval;
- $R_b(a, j, b)$ is the conditional recovery rate for reference name a in the time interval $(T_j, T_{j+1}]$, conditional to the conditioning process visiting the bond b in the same time interval;
- $R_{\downarrow}(a, j, v)$ is the conditional recovery rate for reference name a in the time interval $(T_j, T_{j+1}]$, conditional to the conditioning process visiting the vertex v at time T_j and then jumping from there to the lowest vertex;
- $L_a(a, j)$ is the expected loss for name a in the time interval $(T_j, T_{j+1}]$;
- $L_{av}(a, j, v)$ is the expected loss for name a in the time interval $(T_j, T_{j+1}]$, conditional to the vertex v being visited by the single name conditioning process at time T_j;
- $N_z^1 + 1$ is the number of discretization point for the single name loss distribution, so that this is parameterized by an integer $z = 0, ... N_z^1$;
- $\overline{F_{va}}(a, j, v, z)$ is the loss distribution function as indexed by the variable z for name a in the time interval $(T_j, T_{j+1}]$, conditional to the vertex v being visited by the single name conditioning process at time T_j;
- $F_{ba}(a, j, b, z)$ is the differential of the loss distribution function as indexed by the variable z for name a in the time interval $(T_j, T_{j+1}]$, conditional to the bond b being visited by the single name conditioning process at time T_j;
- $F_{ja}(a, j, v, z)$ is the differential of the loss distribution function as indexed by the variable z for name a in the time interval $(T_j, T_{j+1}]$, conditional to the vertex v being visited by the single name conditioning process at time T_j and conditioned to a jump to the lowest vertex occurring in the same time interval;
- $L_s(\texttt{sec}, j)$ is the expected loss for the sector $\texttt{sec}$ in the time interval $(T_j, T_{j+1}]$;
- $L_{sv}(\texttt{sec}, j)$ is the conditional expected loss for the sector $\texttt{sec}$ in the time interval $(T_j, T_{j+1}]$, conditional to the sector process visiting the vertex v at time T_j;
- $L(j)$ is the global basket expected loss in the time interval $(T_j, T_{j+1}]$.

Firstly, using the results in the previous section on dynamic conditioning and the single name initialization, we set

$$D_{ba}^{\text{eqb}}(a, j, b) = D_{by}^{\text{eqb}}(y_0(a), j, b),$$
$$D_{va}^{\text{eqj}}(a, j, v) = D_{va}^{\text{eqj}}(y_0(a), j, v),$$
$$\overline{D_{va}^{\text{eq}}}(a, j, v) = \overline{D_{va}^{\text{eq}}}(y_0(a), j, v).$$

(77)

The calculation of the quantities above is by iteration in j. Suppose that $j > 0$ and let v_0 be a vertex at time T_j. There is at least one bond terminating at v_0, let's denote with b_1 the lowest one and let $v_1 = V_{\leftarrow}(b_1)$. If there is a second bond with $V_{\rightarrow}(j, b) = v_0$, then let us denote it with b_2 and let $v_2 = V_{\leftarrow}(b_1)$. Let $1_{b_2}(v_0, j)$ denote a function which equals one in case the bond b_2 exists and zero otherwise.

Let us introduce the constant

$$N(v_0) = P_b(j-1, b_1) + 1_{b_2}(v_0, j)P_b(j-1, b_1) + \delta_{v_0, v_{\min}(j)} \sum_{v_3 = v_{\min}(j-1)}^{v_{\max}(j-1)} P_\downarrow(j-1, v_3).$$

(78)

Let us set

$$\begin{aligned}
\overline{D_{va}}(a, j, v_0) = {}& N(v_0)^{-1}P_b(j-1, b_1) \cdot \left[\overline{D_{va}}(a, j-1, v_1) + D_{ba}(a, j-1, b_1)\right] \\
& + 1_{b_2}(v_0, j)N(v_0)^{-1}P_b(j-1, b_2) \cdot \left[\overline{D_{va}}(a, j-1, v_2) + D_{ba}(a, j-1, b_2)\right] \\
& + N(v_0)^{-1}\delta_{v_0, v_{\min}(j-1)} \sum_{v_3 = v_{\min}(j-1)}^{v_{\max}(j-1)} P_\downarrow(j-1, v_3) \cdot \left(\overline{D_{va}}(a, j-1, v_3) + D_{ja}(a, j-1, v_3)\right)
\end{aligned}$$

(79)

Finally

$$(80) \qquad \overline{D_a}(a, j) = \sum_{a=1..N_a} \sum_{v_0 = v_{\min}(j)}^{v_{\max}(j)} P_v(j, v_0) \cdot \overline{D_{va}}(a, j, v_0).$$

As we iterate through the bonds, we also calibrate to the market CDS curves. The general expression for $D_{ba}(a, j, b)$ is

(81)

$$D_{ba}(a, j, b) = (1 - \overline{D_{va}}(a, j, V_\leftarrow(j, b)))(q_b(a, j, \bar{b}) + \pi_{ba}^{\text{jtd}}(a, j, b) - \pi_{ba}^{\text{jtd}}(a, j, b)q_b(a, j, \bar{b}))$$

where

$$(82) \qquad q_b(a, j, b) = \frac{D_{ba}^{\text{eqb}}(a, j, b)}{1 - \overline{D_{va}^{\text{eq}}}(a, j, V_\leftarrow(j, b))}.$$

is the conditional probability that a default occurs in the underlying process in the time interval $(T_j, T_{j+1}]$, conditional to no default occurring in the underlying process up to time T_j. The goal of the calibration algorithm is to choose $\pi_{ba}^{\text{jtd}}(a, j, b)$ so to ensure that the market CDS spread curves for each reference name are precisely reobtained. To be more precise, let us also set

$$(83) \qquad q_\downarrow(a, j, v) = \frac{D_{va}^{\text{eqj}}(a, j, v)}{1 - \overline{D_{va}^{\text{eq}}}(a, j, v)}$$

and

$$(84) \qquad D_{ja}(a, j, v) = \left(1 - \overline{D_{va}}(a, j, V_\leftarrow(j, b))\right) \cdot q_\downarrow(a, j, v).$$

We have that

$$(85) \quad D_a(a, j) = \sum_{b=b_{\min}(j)}^{b_{\max}(j)} P_b(j, b) \cdot D_{ba}(a, j, b) + \sum_{v=v_{\min}(j)}^{v_{\max}(j)} P_v(j, v) \cdot D_{ja}(a, j, v).$$

The goal of the calibration algorithm is to choose $\pi_{ba}^{\text{jtd}}(a, j, b)$ in such a way that

$$(86) \qquad D_a(a, j) = \lambda_a(a, j).$$

Let $j \geq 0$ and consider the conditional probability of equity driven default in the interval $(T_j, T_{j+1}]$, conditional to no equity driven default having occurred prior to time T_j, i.e.

$$
D_a^{eq}(a, j) = \sum_{b=b_{\min}(j)}^{b_{\max}(j)} P_b(j, b) \cdot (1 - \overline{D_{va}}(a, j, V_\leftarrow(j, b))) \cdot q_b(a, j, b)
$$
$$
(87) \qquad + \sum_{v=v_{\min}(j)}^{v_{\max}(j)} P_\downarrow(j, v) \cdot (1 - \overline{D_{va}}(a, j, v)) \cdot q_\downarrow(a, j, v).
$$

We start adding jumps from the bottom vertex $b = b_{\min}(j)$ and proceed upward by induction. At each step, if we are looking at the bond of index $b \geq b_{\min}(j)$, we first consider setting

$$
(88) \qquad \pi_{ba}^{\text{jtd}}(a, j, b) = 1.
$$

Assuming that the conditional jump probability is 1, as in (88), we find

$$
(89) \qquad D_{ba}(a, j, b) = 1 - \overline{D_{va}}(a, j, V_\leftarrow(j, b)).
$$

Let us consider the partial sum

$$
(90) \qquad \Delta(a, j, b) = D_a^{eq}(a, j) + \sum_{b_1=b_{\min}(j)}^{b} P_b(j, b) \cdot D_{ba}(a, j, b_1).
$$

Let $\bar{b}$ be the first bond encountered proceeding from $b_{\min}(j)$ such that the following inequality is satisfied:

$$
(91) \qquad \Delta(a, j, \bar{b}) \leq \lambda_a(a, j).
$$

For all $b < \bar{b}$, equation (88) is retained as valid and one iterates the construction to the next bond. Otherwise, one needs to fine-tune the choice of conditional jump probability to obtain the correct fit to the implied probability of default $\lambda_a(a, j)$. We set

$$
D_{ba}(a, j, \bar{b}) = P_b(j, \bar{b})^{-1}(\lambda_a(a, j) - \Delta(a, j, \bar{b})) + (1 - \overline{D_{va}}(a, j, V_\leftarrow(j, \bar{b}))) \cdot q_b(a, j, \bar{b}),
$$
$$
(92)
$$

$$
(93) \qquad \pi_{ba}^{\text{jtd}}(a, j, \bar{b}) = (1 - q_b(a, j, \bar{b}))^{-1} \left[\frac{D_{ba}(a, j, \bar{b})}{1 - \overline{D_{va}}(a, j, V_\leftarrow(j, \bar{b}))} - q_b(a, j, \bar{b}) \right],
$$

and

$$
(94) \qquad D_{ba}^{\text{jtd}}(a, j, \bar{b}) = \pi_{ba}^{\text{jtd}}(a, j, \bar{b}) \cdot \left(1 - \overline{D_{va}}(a, j, V_\leftarrow(j, \bar{b})) \right).
$$

Having done this, for larger values of $b > \bar{b}$, it is no longer required to add jump to default amplitudes to calibrate and one can set

$$
(95) \qquad \pi_{ba}^{\text{jtd}}(a, j, b) = 0, \qquad\qquad D_{ba}^{\text{jtd}}(a, j, b) = 0
$$

and

$$
(96) \qquad D_{ba}(a, j, b) = (1 - \overline{D_{va}}(a, j, V_\leftarrow(j, b))) q_b(a, j, \bar{b}).
$$

Next, one needs to firm up the conditional recovery rates $R_b(a, j, b)$ and $R_\downarrow(a, j, v)$. One can simply set

$$
(97) \qquad R_b(a, j, b) = R_\downarrow(a, j, v) = R(a, j).
$$

Otherwise, one can model stochastic recovery rates by having $R_b(a, j, b)$ depend on b and $R_\downarrow(a, j, v)$ depend on v. In this case, several strategies are possible and we won't discuss in further detail here except for saying that an acceptable choice will satisfy the following constraint:

$$\sum_{b=b_{\min}(j)}^{b_{\max}(j)} P_b(j, b) \cdot D_{ba}(a, j, b) \cdot R_b(a, j, b) + \sum_{v=v_{\min}(j)}^{v_{\max}(j)} P_v(j, v) \cdot D_{ja}(a, j, v) \cdot R_\downarrow(a, j, v) = R(a, j).$$

(98)

Finally, we consider cumulative losses on vertices v at time T_{j+1}. If $j = 0$, we set

(99) $$\overline{F_{va}}(a, j = 0, v = 0, z) = \delta_{z,0}.$$

Let b_1 be the lowest bond such that $V_\rightarrow(j, b_1) = v$. Let $1_{b_2}(v_0, j)$ be a function equal to one in case there exists a second bond $b_2 \neq b_1$ with $V_\rightarrow(j, b_2) = v$. Let the function $N(v_0)$ be defined as in equation (78). We have that

$$\overline{F_{va}}(a, j + 1, v_0, z) = N(v_0)^{-1} P_b(j, b_1) \cdot (\overline{F_{va}}(a, j, v_1, z) + F_{ba}(a, j, b_1, z))$$
$$+ N(v_0)^{-1} P_b(j, b_1) \cdot (\overline{F_{va}}(a, j, v_2, z) + F_{ba}(a, j, b_2, z))$$
$$+ N(v_0)^{-1} \delta_{v_0, v_{\min}(j)} \sum_{v_3 = v_{\min}(j)}^{v_{\max}(j)} P_\downarrow(j, v_3) \cdot (\overline{F_{va}}(a, j, v_3, z) + F_{ja}(a, j, v_3, z))$$

(100)

Here

$$F_{ba}(a, j, b, z) = D_{ba}(a, j, b) \cdot \left(-\delta_{z0} + (z_2 - \ell_b(a, j, b))\delta_{zz_1} + (\ell_b(a, j, b) - z_1)\delta_{zz_2} \right),$$
(101)

where

$$\ell_b(a, j, b) = N_z^1 \cdot (1 - R_b(a, j, b))$$
$$z_1 = \text{floor}(\ell_b(a, j, b))$$
(102) $$z_2 = \min(z_1 + 1, N_z^1).$$

Furthermore

$$F_{ja}(a, j, v, z) = D_{ja}(a, j, v) \cdot \left(-\delta_{z0} + (z_2 - \ell_\downarrow(a, j, v))\delta_{zz_1} + (\ell_\downarrow(a, j, v) - z_1)\delta_{zz_2} \right),$$
(103)

where

$$\ell_\downarrow(a, j, b) = N_z^1 \cdot (1 - R_\downarrow(a, j, v))$$
$$z_1 = \text{floor}(\ell_\downarrow(a, j, v))$$
(104) $$z_2 = \min(z_1 + 1, N_z^1).$$

Finally, expected cumulative losses are given as follows:

$$L_{av}(a, j, v) = \frac{1}{N_z^1} \sum_{z=0}^{N_z^1} \overline{F_{va}}(a, j, v, z) \cdot z$$

$$L_a(a, j) = \sum_{v=v_{\min}(j)}^{v_{\max}(j)} P_v(j)(v) L_{av}(a, j, v)$$

$$L_{sv}(\mathtt{sec}, j, v) = \sum_{a:\mathtt{sec}(a)=\mathtt{sec}} L_{av}(a, j, v)$$

$$L_s(\mathtt{sec}, j) = \sum_{a:\mathtt{sec}(a)=\mathtt{sec}} L_a(a, j)$$

$$L(j) = \sum_s L_s(\mathtt{sec}, j).$$

(105)

6. Loss Distribution and Tranche Pricing

Let us introduce the following notations:

- $N_z = N_a \cdot N_z^1$;
- $Q_{vv}^1(a, j, v_1, v_2)$ is the joint probability that the conditioning process for name a visits the vertex v_1 while the corresponding sector conditioning process visits vertex v_2;
- $Q_{vv}^2(\mathtt{sec}, j, v_2, v_3)$ is the joint probability that the conditioning process for the sector $\mathtt{sec}$ visits the vertex v_2 while the corresponding global conditioning process visits vertex v_3;
- $\overline{\ell}(j, z)$ is the (unconditional) cumulative loss distribution up to time T_j, where $j = 0, ... N_j$ and $z = 0, ... N_z$;
- $\overline{\ell_{12}}(a, j, v_2, z)$ is the conditional cumulative loss distribution for name a at time T_j, conditioned to the sector conditioning process visiting the vertex v_2 at time T_j and $z = 0, ... N_z$;
- $\overline{\ell_{22}}(\mathtt{sec}, j, v_2, z)$ is the conditional cumulative loss distribution for sector $\mathtt{sec}$ at time T_j, conditioned to the sector conditioning process visiting the vertex v_2 at time T_j and $z = 0, ... N_z$;
- $\overline{\ell_{23}}(\mathtt{sec}, j, v_3, z)$ is the conditional cumulative loss distribution for sector $\mathtt{sec}$ at time T_j, conditioned to the global conditioning process visiting the vertex v_3 at time T_j and $z = 0, ... N_z$;
- $\widehat{\overline{\ell_{12}}}(\mathtt{sec}, j, v_3, k)$ is the Fourier transform in the z variable of the function $\overline{\ell_{23}}(\mathtt{sec}, j, v_3, z)$ and $z = 0, ... N_z$;
- $\overline{\ell_{33}}(j, v_3, z)$ is the conditional cumulative loss distribution for the portfolio at time T_j, conditioned to the global conditioning process visiting the vertex v_3 at time T_j and $z = 0, ... N_z$;
- $\widehat{\overline{\ell_{12}}}(j, v_3, k)$ is the Fourier transform in the z variable of the function $\overline{\ell_{33}}(j, v_3, z)$ and $z = 0, ... N_z$;
- $P_{\mathtt{tr}}^{\mathtt{d}}$ is the present value at initial time of the default leg for the tranche $\mathtt{tr}$ including only defaults up to time T_j;

- $P^{\mathtt{d}}$ is the present value at initial time of the default leg for the entire basket including only defaults up to time T_j;
- $P^{\mathtt{ra}}_{\mathtt{tr}}$ is the present value at initial time of the risky annuity for the tranche $\mathtt{tr}$ including only payments up to time T_j;
- $P^{\mathtt{ra}}$ is the present value at initial time of the risky annuity for the entire basket including only payments up to time T_j;
- $s_{\mathtt{tr}}$ is the equilibrium spread at initial time for the tranche $\mathtt{tr}$ with maturity T_j;
- $a_{\mathtt{tr}}$ is the percentage attachment point of the tranche tr;
- $d_{\mathtt{tr}}$ is the percentage detachment point of the tranche tr;
- $l_{\mathtt{tr}}$ is the attachment point of the tranche tr approximated as an integer and supposing that the loss amount for each individual name is an integer in the interval $[0, ..N_z]$;
- $u_{\mathtt{tr}}$ is the detachment point of the tranche tr approximated as an integer and supposing that the loss amount for each individual name is an integer in the interval $[0, ..N_z]$;
- $f_{\mathtt{tr}}$ is the equilibrium upfront fee at initial time for the tranche $\mathtt{tr}$ with maturity T_j;
- $\ell(j, z)$ is the differential of the (unconditional) cumulative loss distribution up to time T_j, where $j = 0, ...N_j$ and $z = 0, ...N_z$.

The construction proceeds iteratively in j, starting from $j = 0$ and arriving to $j = N_j$. If $j = 0$, the initialization conditions are

$$(106) \qquad \overline{\ell_{12}}(a, j = 0, v_2 = 0, z) = \delta_{z0} \qquad\qquad \forall a,$$

$$(107) \qquad \overline{\ell_{22}}(\mathtt{sec}, j = 0, v_2 = 0, z) = \delta_{z0} \qquad\qquad \forall\mathtt{sec},$$

$$(108) \qquad \overline{\ell_{23}}(\mathtt{sec}, j = 0, v_3 = 0, z) = \delta_{z0} \qquad\qquad \forall\mathtt{sec},$$

$$(109) \qquad \overline{\ell_{33}}(j = 0, v_3 = 0, z) = \delta_{z0},$$

$$(110) \qquad \overline{\ell}(j = 0, z) = \delta_{z0}.$$

Assume that $j > 0$. For all reference names a, we have

$$(111) \qquad \overline{\ell_{12}}(a, j, v_2, z) = \sum_{v_1 = v_{\min}(j)}^{v_{\max}(j)} P_v(j, v_2)^{-1} Q^1_{vv}(a, j, v_1, v_2) \overline{F_{va}}(a, j, v_1, z).$$

The function $\overline{F_{va}}(a, j, v_1, z)$ was defined and discussed in Section 6. The sector cumulative distribution is defined as a convolution product in the z variable, i.e.

$$(112) \qquad \overline{\ell_{22}}(\mathtt{sec}, j = 0, v_2 = 0, \cdot) = \overline{\ell_{12}}(a_1, j, v_2, \cdot) * .... * \overline{\ell_{12}}(a_n, j, v_2, \cdot)$$

where $(a_1, ...a_n)$ are the reference names in the sector $\mathtt{sec}$. We also have that

$$(113) \quad \overline{\ell_{23}}(\mathtt{sec}, j, v_3, z) = \sum_{v_2 = v_{\min}(j)}^{v_{\max}(j)} P_v(j, v_3)^{-1} Q^2_{vv}(\mathtt{sec}, j, v_2, v_3) \overline{\ell_{22}}(\mathtt{sec}, j, v_2, z).$$

Evaluating the Fourier transforms

$$(114) \qquad\qquad \widehat{\overline{\ell_{12}}}(\mathtt{sec}, j, v_3, k) = \mathcal{F}\big[\overline{\ell_{23}}(\mathtt{sec}, j, v_3, \cdot)\big](k),$$

we find

$$(115) \qquad \widehat{\overline{\ell_{12}}}(j, v_3, k) = \prod_{\mathtt{sec}} \widehat{\overline{\ell_{12}}}(\mathtt{sec}, j, v_3, k).$$

The inverse Fourier transform

$$(116) \qquad \overline{\ell_{33}}(j, v_3, z) = \mathcal{F}^{-1}\big[\widehat{\overline{\ell_{12}}}(j, v_3, \cdot)\big](z)$$

gives the loss distribution conditional to the global conditioning process visiting the vertex v_3 at time T_j. Finally, the cumulative loss distribution is given by

$$(117) \qquad \overline{\ell}(j, z) = \sum_{v_3 = v_{\min}(j)}^{v_{\max}(j)} P_v(j, v_3) \cdot \overline{\ell_{33}}(v_3, z).$$

Given the cumulative loss distribution, one can evaluate tranche spreads. The attachment and detachment points are given by

$$l_{\mathtt{tr}}(tr) = \mathtt{floor}(N_z \cdot a_{\mathtt{tr}}(tr) + 1),$$
$$(118) \qquad u_{\mathtt{tr}}(tr) = \mathtt{floor}(N_z \cdot d_{\mathtt{tr}}(tr)).$$

The differential loss distribution is defined as follows:

$$(119) \qquad \ell(j, z) = \overline{\ell}(j+1, z) - \overline{\ell}(j, z).$$

for all $j \geq 0$. Furthermore, we have that

$$P_{\mathtt{tr}}^{\mathtt{d}}(j, \mathtt{tr}) = P_{\mathtt{tr}}^{\mathtt{d}}(j-1, \mathtt{tr}) + Z(T_j) \sum_{z = l_{\mathtt{tr}}(\mathtt{tr})}^{N_z - 1} \ell(j, z) \cdot \min(z - l_{\mathtt{tr}}(\mathtt{tr}) + 1, u_{\mathtt{tr}}(\mathtt{tr}) - l_{\mathtt{tr}}(\mathtt{tr}) + 1).$$

(120)

Furthermore

$$P_{\mathtt{tr}}^{\mathtt{ra}}(j, \mathtt{tr}) = P_{\mathtt{tr}}^{\mathtt{ra}}(j-1, \mathtt{tr}) + Z(T_j)\Delta T\Bigg[\sum_{z=0}^{l_{\mathtt{tr}}(\mathtt{tr})-1} (u_{\mathtt{tr}}(\mathtt{tr}) + 1 - l_{\mathtt{tr}}(\mathtt{tr})) \cdot \overline{\ell}(j, z)$$
$$+ \sum_{z = l_{\mathtt{tr}}(\mathtt{tr})}^{N_z - 1} \max(u_{\mathtt{tr}}(\mathtt{tr}) - z, 0) \cdot \overline{\ell}(j, z)\Bigg].$$

(121)

If the tranche trades with no upfront fee, the spread is given by

$$(122) \qquad s_{\mathtt{tr}}(j, \mathtt{tr}) = \frac{P_{\mathtt{tr}}^{\mathtt{d}}(j, \mathtt{tr})}{P_{\mathtt{tr}}^{\mathtt{ra}}(j, \mathtt{tr})}.$$

If instead the tranche trades with a fixed running spread $s_0(\mathtt{tr})$, the upfront fee is given by

$$(123) \qquad f_{\mathtt{tr}}(j, \mathtt{tr}) = \frac{P_{\mathtt{tr}}^{\mathtt{d}}(j, \mathtt{tr}) - s_0(\mathtt{tr}) \cdot P_{\mathtt{tr}}^{\mathtt{ra}}(j, \mathtt{tr})}{u_{\mathtt{tr}}(tr) - l_{\mathtt{tr}}(\mathtt{tr}) + 1}.$$

As a check for numerical implementations, it is useful to ensure that the following sum rule is satisfied:

$$(124) \qquad \sum_{\mathtt{tr}} P_{\mathtt{tr}}^{\mathtt{d}}(j, \mathtt{tr}) = \sum_{j'=0}^{j} \sum_{a} Z(j') \cdot (L_a(a, j'+1) - L_a(a, j')),$$

whenever the summation in the right hand side extends over any set of tranches covering the entire capital structure without overlaps.

7. Benchmarks and Datasets

We present results referring to the on-the-run investment grade CDX index tranches as calibrated on April 2006, March 2007, October 2007 and March 2008.

Task	CPU Time in seconds	Memory allocated in MB
Preprocessing	137.57	308
Single Name Calibration	2.69	77
CDO Tranche Pricing	8.63	181

Table 1. Execution times on a single processor Xeon machine, 2 GHz, with a nVidia Tesla GPU coprocessor.

We adopted the simplifying assumption that all bond correlations and all jump correlations are the same. This assumption can obviously be refined at no computational cost by assuming instead that correlation depends on the reference name and sector. This hypothesis however allows one to economize on the number of free parameters. We further assume that the two term structures are constant for the April 2006 and March 2007 datasets while a mild time dependence is needed for the October 2007 and March 2008 datasets. In this case, we assume that the term structure of jump correlation is inverted and linear. (Curiously, an inverted term structure for correlation is also observed when parameterizing tranche prices with base correlation). A further free parameter is given by $q(\downarrow)$, while $q(\nearrow)$ and and $q(\searrow)$ are assumed to be equal. See Table 2. Hence, the number of free parameters that we allow ourselves is three for the April 2006 and March 2007 datasets and is equal to four in the case of the October 2007 and March 2008 datasets.

Dataset	$q(\downarrow)$	$q(\nearrow)$	$q(\searrow)$	$\rho_b(0)$	$\rho_b(40)$	$\rho_j(0)$	$\rho_j(40)$
April 2006	0.04%	49.98%	49.98%	65%	65%	85%	85%
March 2007	0.04%	49.98%	49.98%	50%	50%	95%	95%
October 2007	0.20%	49.80%	49.80%	65%	65%	95%	80%
March 2008	0.24%	49.88%	49.88%	95%	95%	80%	99.5%

Table 2. Correlation parameters for the three datasets.

The loss distributions are given in Fig. 7, 8, 9 and 10. The equity upfront fees are in Fig. 11, 12, 13 and 14. The mezzanine tranche spreads are in Fig. 15, 16, 17 and 18. The senior tranche spreads are in Fig. 19, 20, 21 and 22. These graphs showcase the use of the model. The calibration to the index is perfect and the equity and senior tranches are fitted well. The 30-100 super-senior tranche is fitted well only in the most recent datasets. The fit to the mezzanine tranches, which are often the most difficult to price, is not of similar quality. The 2007 datasets show that the market spreads for mezzanine tranches were consistently below the model spreads across all the mezzanine capital structure. Since the model is arbitrage free, this is an indication of mispricing and approximate arbitrage. The march 2008 dataset shows that this arbitrage opportunity has been eliminated and the term structures

of the mezzanine tranches are correctly reproduced by the model. However, the supersenior tranche spreads appear excessive and the short term spreads are also biased downward. Due to the scarce market liquidity and high volatility in recent times, we find that the quality of the fit is acceptable and the model is a useful indicator for price dislocations and a predictor of market adjustments.

8. Conclusions

In this paper we carry forward our work on dynamic conditioning and CDO modeling. We introduce for the first time conditioning trees which are recombining binomial trees also admitting jumps. This technique allows one to model separately spread correlations and default correlations. The former is important to understand the lower portions of the capital structure, the latter matters more for the senior tranches. Here we describe the model in much further detail than before attempted and discuss also a version which is of very efficient numerical implementation.

References

Albanese, C. (2006). Operator Methods, Abelian Path Dependents and Dynamic Conditioning. *preprint*.

Albanese, C. (2007). Kernel Convergence Estimates for Diffusions with Continuous Coefficients. *arXiv:0711.0132v1 [math.NA]*.

Albanese, C. and A. Osseiran (2007). Moment Methods for Exotic Volatility Derivatives. *preprint*.

Albanese, C. and A. Vidler (2006). A Structural Model for Credit-Equity Derivatives and Bespoke CDOs. *Willmott Magazine*.

Albanese, C. and O. Chen (2004). Implied migration rates from credit barrier model. *The Journal of Banking and Finance, to appear*.

Albanese, C. and O. Chen (2005a). Credit barrier models in a discrete framework. *Contemporary Mathematics 351, Mathematical Finance* pp. 1–11.

Albanese, C. and O. Chen (2005b). Discrete credit barrier models. *Quantitative Finance* **5**, 247–256.

Albanese, C., J. Campolieti, O. Chen and A. Zavidonov (2003). Credit barrier models. *Risk* **16**(6), 109–113.

Albanese, C., O. Chen, A. Dalessandro and A. Vidler (2005-2006). Dynamic Credit Correlation Modelling. *preprint*.

Andersen, Leif and Jakob Sidenius (2004). Extensions to the gaussian copula: random recovery and random factor loadings. *Journal of Credit Risk* **1**, 1:29.

di Graziano, G. and C. Rogers (2006). A Dynamic Approach to the Modelling of Correlation Credit Derivatives Using Markov Chains. *preprint, Cambridge University*.

Duffie, D., J. Pan and K. Singleton (2000). Transform analysis and asset pricing for affine jump-diffusions. *Econometrica*.

Duffie, Darrell, Andreas Eckner, Guillaume Horel and Leandro Saita (2006). Frailty Correlated Default. *preprint, Stanford University*.

Giesecke, Kay and Lisa Goldberg (2005). A top down approach to multi-name credit. *Working paper, Cornell University*.

Hull, John and Alan White (2003). Valuation of a cdo and an n-th to-default cds without monte carlo simulation. *Working paper, University of Toronto*.

Joshi, Mark S. and Alan Stacey (2005). Intensity gamma: a new approach to pricing portfolio credit derivatives. *Working paper, Royal Bank of Scotland*.

Li, David. X. (2000). On default correlation: A copula function approach. *working paper 99-07, Risk Metrics Group*.

Lucas, A., P. Klaassen, P. Spreij and S. Staetmans (2001). An analytic approach to credit risk of large corporate bond and loan portfolios. *Journal of Banking and Finance* **9**, 1635–1664.

O'Kane, D. and M. Livesey (2004). Base correlation explained. *QCR Quarterly Q3/4, Lehman Brothers Fixed Income Quantitative Research*.

Schonbucher, P. (2006). Portfolio losses and the term structure of loss transition rates: a new
methodology for the pricing of portfolio credit derivatives. *Working paper, ETHZ.*
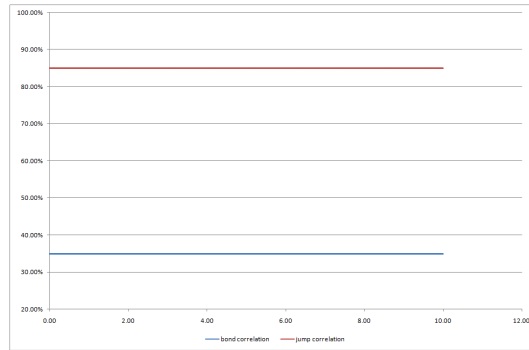
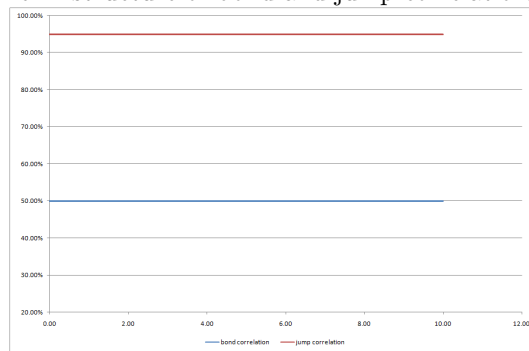FIGURE 3. Term structure of bond and jump correlations, April 2006.

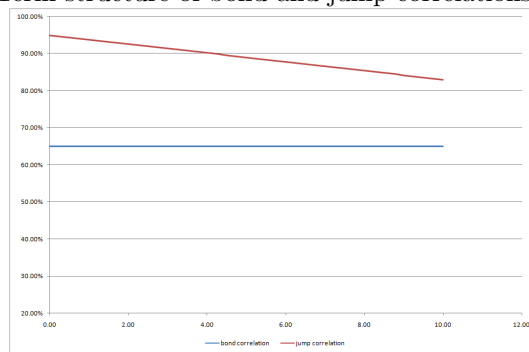FIGURE 4. Term structure of bond and jump correlations, March 2007.

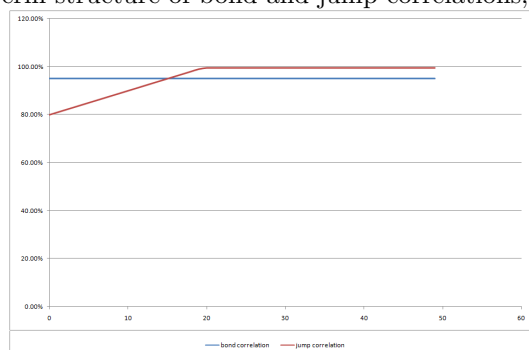FIGURE 5. Term structure of bond and jump correlations, October 2007.

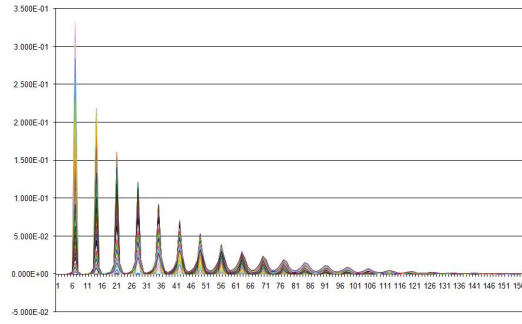FIGURE 6. Term structure of bond and jump correlations, March 2008.

FIGURE 7. Term structure of cumulative loss distributions, April 2006.
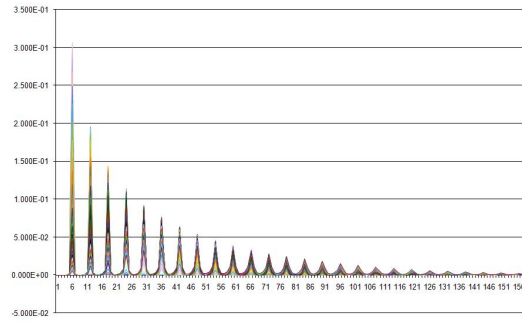
FIGURE 8. Term structure of cumulative loss distributions, March 2007.
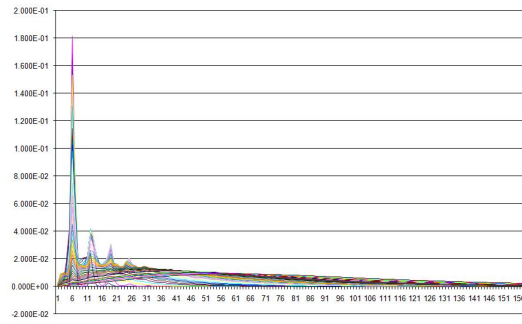
FIGURE 9. Term structure of cumulative loss distributions, October 2007.
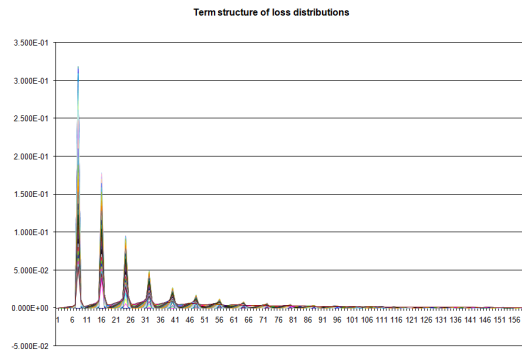
FIGURE 10. Term structure of cumulative loss distributions, March 2008.
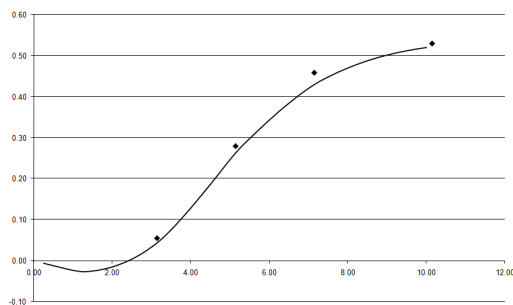
FIGURE 11. 03 equity tranche upfront fee (with 500bp running spread), April 2006.
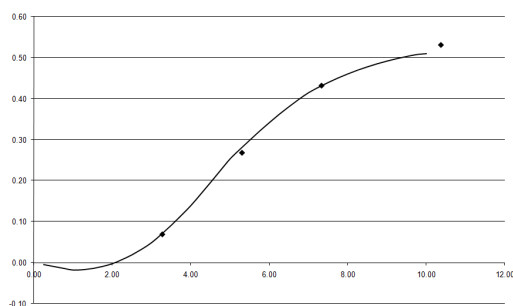
FIGURE 12. 03 equity tranche upfront fee (with 500bp running spread), March 2007.
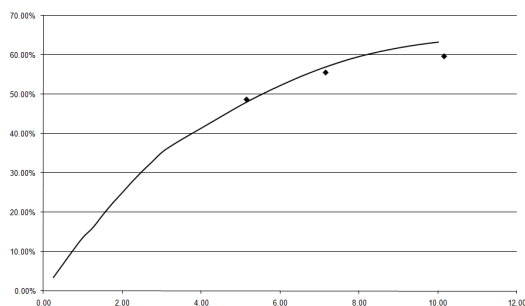
FIGURE 13. 03 equity tranche upfront fee (with 500bp running spread), October 2007.
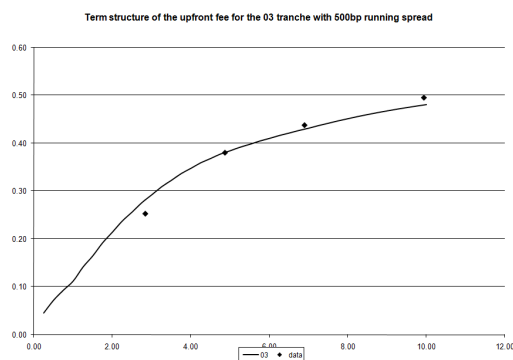
FIGURE 14. 03 equity tranche upfront fee (with 500bp running
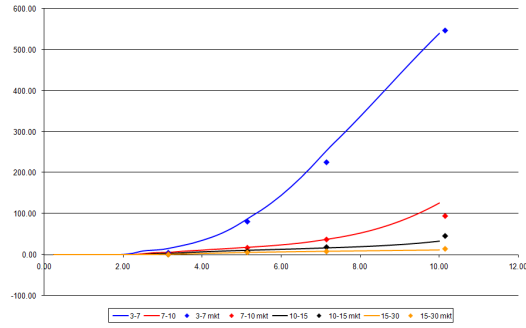
FIGURE 15. Mezzanine tranche spreads, April 2006.
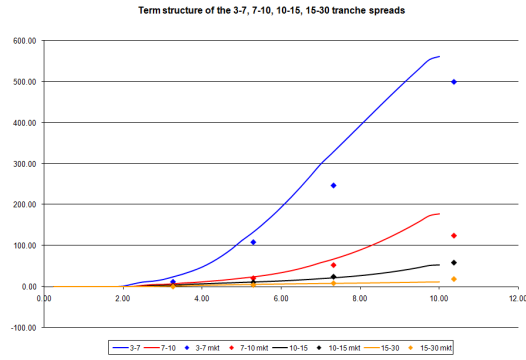
FIGURE 16. Mezzanine tranche spreads, March 2007.
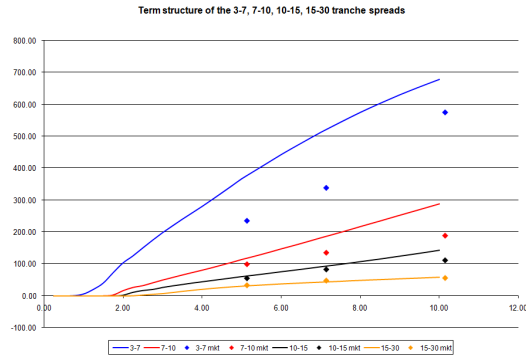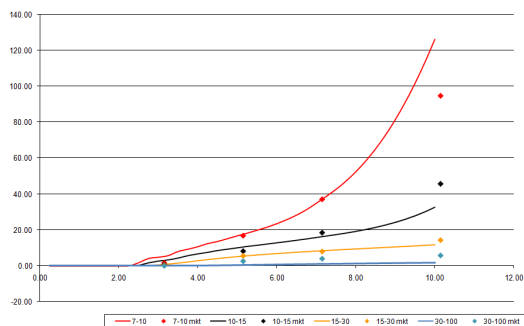

FIGURE 17. Mezzanine tranche spreads, October 2007.
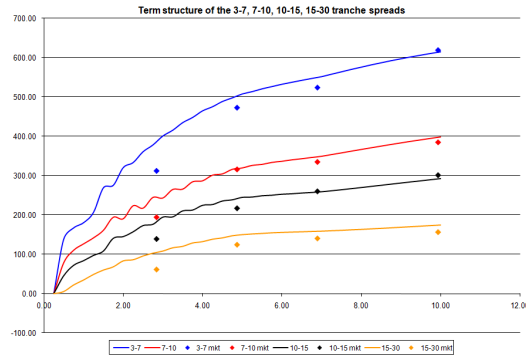
FIGURE 18. Mezzanine tranche spreads, March 2008.

Figure 19. Senior tranche spreads, April 2006.
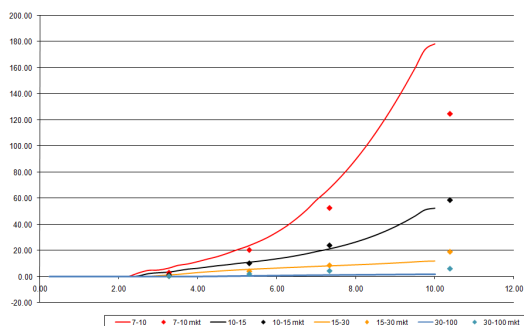

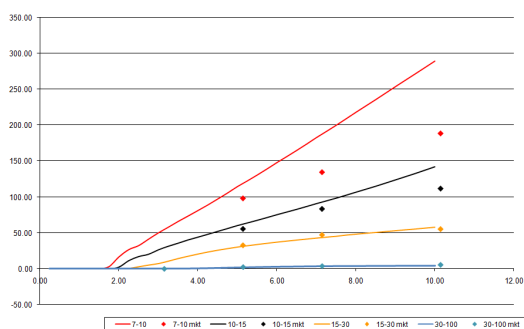Figure 20. Senior tranche spreads, March 2007.
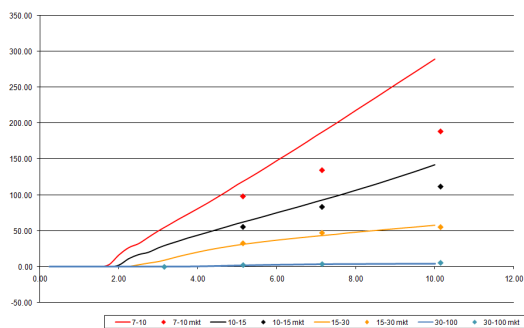

Figure 21. Senior tranche spreads, October 2007.

Figure 22. Senior tranche spreads, March 2008.